



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих
комп'ютерних систем**

Розрахунково-графічна робота

з дисципліни **Бази даних і засоби управління**

*на тему: “ Створення додатку бази даних, орієнтованого на
взаємодію з СУБД PostgreSQL ”*

Виконав: студент III курсу

ФПМ групи КВ-23

Мамчиць Микола

Посилання на репозиторій: <https://github.com/Mykola6117/RGR-DB>

телеграм: @belqa

Перевірив:

Київ – 2024

Завдання на розрахункову роботу:

- 1.Реалізувати функції перегляду, внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі №1, засобами консольного інтерфейсу.
- 2.Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.
- 3.Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.
- 4.Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

Короткий опис бази даних

Перелік сутностей та їх призначення:

1.Лікар:

Призначення: представляє лікарів, що ведуть медичні історії та ставлять діагнози.

Основні атрибути:

Лікар_id: унікальний ідентифікатор лікаря.

П.І.Б: повне ім'я лікаря.

Спеціалізація: спеціалізація лікаря.

2.Історія_Хвороби:

Призначення: зберігає інформацію про медичні записи пацієнтів, їх діагнози, а також лікаря, що проводив обстеження.

Основні атрибути:

Історія_хвороби_id: унікальний ідентифікатор медичної історії.

Назва хвороби : назва діагнозу.

Опис: опис діагнозу або лікування.

3.Пацієнт:

Призначення: представляє пацієнтів, які проходять медичне обстеження та лікування.

Основні атрибути:

Пацієнт_id: унікальний ідентифікатор пацієнта.

П.І.Б: повне ім'я пацієнта.

стать: стать пацієнта.

Дата народження: дата народження пацієнта.

4.Рецепт:

Призначення: представляє рецепти на ліки, виписані лікарями для пацієнтів у межах їх медичної історії.

Основні атрибути:

Рецепт_id: унікальний ідентифікатор рецепту.

Назва препарату: назва виписаних ліків.

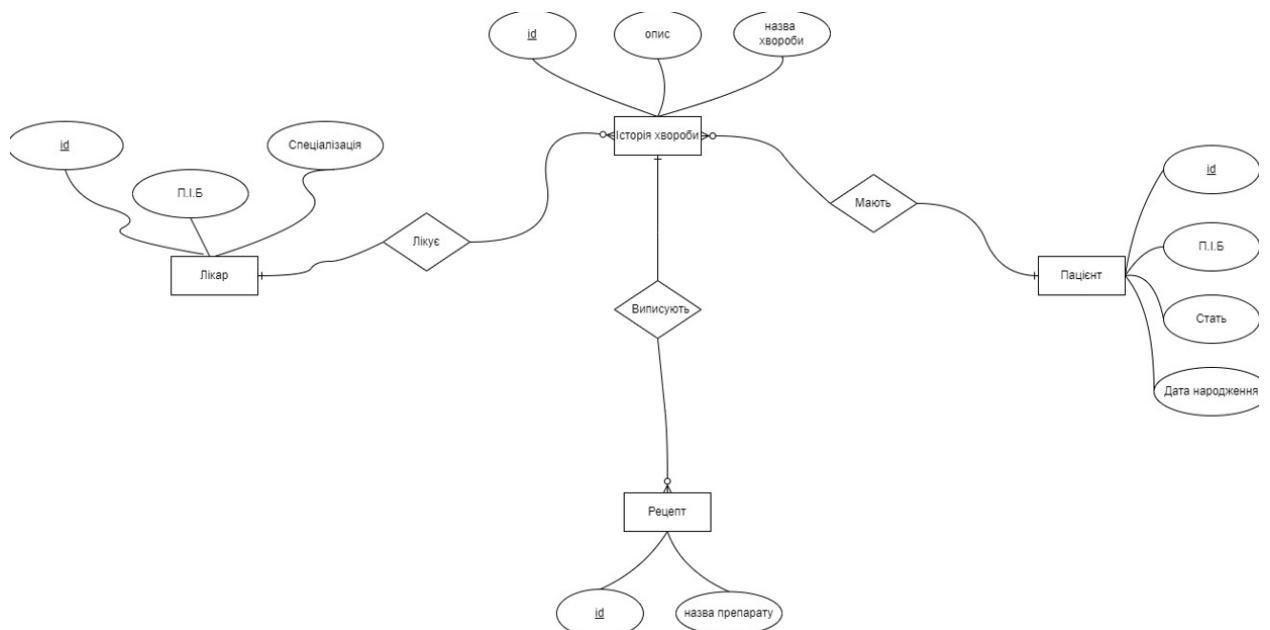
Опис зв'язків між сутностями предметної області

Сутність “Лікар” має зв'язок 1:N по відношенню до сутності “Історія хвороби” оскільки один лікар може написати багато історій хвороби.

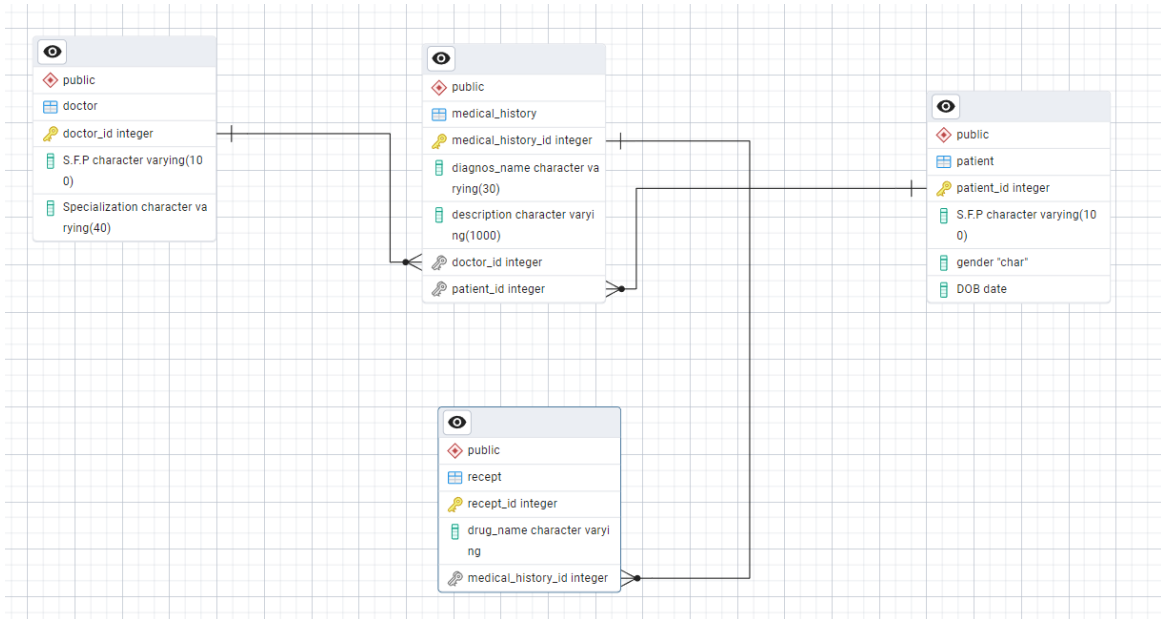
Сутність “Пацієнт” має зв'язок 1:N по відношенню до “Історії хвороби”, оскільки пацієнт може мати безліч історій хвороби.

Сутність “Історія Хвороби” має зв'язок має зв'язок 1:N по відношенню до “Рецепт”, оскільки одна історія хвороби може містити багато рецептів.

ER-діаграма, побудована за нотацією Чена (draw.io)



Логічна модель



Для написання програми була використанна мова програмування Python версії 3.12.7 , з використанням бібліотек psycopg2, time , string , random
Середовище розробки програмного забезпечення – PyCharm Community Edition.

Опис меню програми

Menu:

1. Show table names
2. Show column names of a table
3. Add data to a table
4. Update data in a table
5. Delete data from a table
6. Generate data for a table
7. View data in a table
8. Find medical history by doctor and patient ID
9. Find prescription by medical history ID
10. Exit

1. Show table names

Опис: Показує назви всіх таблиць, що знаходяться в базі даних.

Призначення: Це дозволяє користувачеві отримати список всіх доступних таблиць, з якими можна працювати. Це корисно, щоб зрозуміти, які таблиці є в базі даних і що вони можуть містити.

2. Show column names of a table

Опис: Показує назви всіх стовпців (полів) для вибраної таблиці.

Призначення: Це дає змогу дізнатися структуру таблиці, тобто які саме дані в ній зберігаються. Знання імен стовпців дозволяє зрозуміти, які поля доступні для додавання, оновлення, видалення або перегляду даних.

3. Add data to a table

Опис: Дозволяє додати новий запис (рядок) в обрану таблицю.

Призначення: Ця опція дозволяє користувачеві вносити нові дані в базу даних. Наприклад, додати нового пацієнта, лікаря, медичну історію, рецепт або іншу інформацію, що зберігається в таблицях.

4. Update data in a table

Опис: Дозволяє оновити існуючий запис у вибраній таблиці.

Призначення: Цей пункт меню дає можливість змінювати існуючі дані в базі даних. Наприклад, якщо потрібно оновити контактну інформацію пацієнта чи виправити інформацію про лікаря, можна скористатися цією функцією.

5. Delete data from a table

Опис: Дозволяє видалити один або кілька записів із вибраної таблиці.

Призначення: Видалення даних корисне, коли потрібно прибрати застарілу або помилкову інформацію з бази даних. Наприклад, можна видалити записи про пацієнтів, які більше не обслуговуються, або інші непотрібні дані.

6. Generate data for a table

Опис: Генерує випадкові дані для вибраної таблиці з використанням SQL-запитів PostgreSQL.

Призначення: Ця функція дозволяє автоматично заповнювати таблицю псевдовипадковими даними. Це зручно для тестування і швидкого створення великої кількості даних. Наприклад, можна згенерувати список пацієнтів, лікарів або медичних історій для перевірки функціональності системи.

7. View data in a table

Опис: Показує всі записи у вибраній таблиці.

Призначення: Дозволяє переглядати всі дані, що містяться в конкретній таблиці. Це корисно для швидкого ознайомлення з даними та перевірки їхньої цілісності та актуальності.

8. Find medical history by doctor and patient ID

Опис: Шукає медичну історію в базі даних за ID лікаря та ID пацієнта.

Призначення: Цей пункт дозволяє користувачеві знайти медичні записи, пов'язані з конкретним лікарем та пацієнтом. Це може бути корисно для швидкого доступу до потрібних медичних даних для огляду історії хвороби, результатів обстежень або інших медичних подій.

9. Find prescription by medical history ID

Опис: Знаходить інформацію про призначені ліки за ID медичної історії.

Призначення: Дозволяє знайти всі рецепти, видані на основі певного запису медичної історії. Це корисно для перегляду призначень, що були зроблені лікарем для лікування конкретного випадку або діагнозу.

10. Exit

Опис: Завершує роботу програми.

Призначення: Цей пункт дозволяє користувачеві безпечно закрити програму після завершення роботи з базою даних.

Результати та лістинги

Приклад отримання імен таблиць та імен стовпців

```
Please make a selection: 1
Table names:
recept
doctor
medical_history
patient
Menu:
```

```
Please make a selection: 2
Enter table name: doctor
Column names:
doctor_id
s_f_p
specialization
```

```
SELECT table_name FROM information_schema.tables WHERE
table_schema = 'public'
```

```
SELECT column_name FROM information_schema.columns WHERE
table_name = %s ORDER BY ordinal_position with parameter: doctor
```

Додавання даних у таблицю

```
Please make a selection: 3
Enter table name: doctor
Enter column names (space-separated): doctor_id s_f_p specialization
Enter corresponding values (space-separated): 6 Mamchyts hiryrh
```

INSERT INTO "public"."doctor" (doctor_id, s_f_p, specialization) VALUES (%s, %s, %s) with values: ['6', 'Mamchyts', 'hiryrh']

	doctor_id [PK] integer	s_f_p character varying (100)	specialization character varying (40)
1	1	EM	Neurology
2	2	WE	Orthopedics
3	3	ML	Cardiology
4	4	PQ	Neurology
5	5	VG	Dermatology
6	6	Mamchyts	hiryrh

Перегляд даних у таблиці

```
doctor_id: 2 s_f_p: WE specialization: Orthopedics
doctor_id: 3 s_f_p: ML specialization: Cardiology
doctor_id: 4 s_f_p: PQ specialization: Neurology
doctor_id: 5 s_f_p: VG specialization: Dermatology
doctor_id: 6 s_f_p: Mamchyts specialization: hiryrh
```

Змінна даних

```
Please make a selection: 4
Enter table name: doctor
Enter column names (space-separated) to update: specialization
Enter the ID of the row to update: 6
Enter 1 new values: Ivanov
```

UPDATE doctor SET specialization = %s WHERE doctor_id = %s with parameters: ['Ivanov'], 6

	doctor_id [PK] integer	s_f_p character varying (100)	specialization character varying (40)
1	1	EM	Neurology
2	2	WE	Orthopedics
3	3	ML	Cardiology
4	4	PQ	Neurology
5	5	VG	Dermatology
6	6	Mamchyts	Ivanov

Видалення даних з таблиці:

```
Please make a selection: 5
Enter table name: doctor
Enter the ID of the row to delete: 6
```

Результат після видалення та лог

	doctor_id [PK] integer	s_f_p character varying (100)	specialization character varying (40)
1	1	EM	Neurology
2	2	WE	Orthopedics
3	3	ML	Cardiology
4	4	PQ	Neurology
5	5	VG	Dermatology

DELETE FROM doctor WHERE doctor_id = %s with parameter: 6

Генерація даних в таблицю(2 таблиці по 100000 значень):

```
Please make a selection: 6
Enter table name: doctor
Enter the number of rows to generate: 100000
```

```
Please make a selection: 6
Enter table name: patient
Enter the number of rows to generate: 100000
```

	doctor_id [PK] integer	s_f_p character varying (100)	specialization character varying (40)
5750	5750	LA	Pediatrics
5751	5751	NY	Dermatology
5752	5752	EU	Orthopedics
5753	5753	QX	Dermatology
5754	5754	QI	Neurology
5755	5755	GA	Neurology
5756	5756	GP	Orthopedics
5757	5757	BP	Neurology
5758	5758	PV	Pediatrics
5759	5759	LS	Cardiology
5760	5760	SX	Dermatology
5761	5761	NL	Cardiology
5762	5762	YW	Pediatrics
5763	5763	BJ	Dermatology
5764	5764	HA	Pediatrics
5765	5765	IG	Neurology
5766	5766	XK	Dermatology
5767	5767	JS	Pediatrics
5768	5768	SF	Dermatology
5769	5769	LA	Pediatrics
5770	5770	SA	Cardiology

	patientId [PK] integer	s_f_p character varying (100)	gender character varying	dob date
517	517	VW	Male	1986-03-09
518	518	UK	Male	1951-04-17
519	519	PT	Male	1991-09-19
520	520	NH	Male	1999-09-10
521	521	PB	Female	1969-12-16
522	522	OV	Female	1993-12-24
523	523	JR	Male	1954-05-21
524	524	FG	Female	1979-06-29
525	525	XD	Female	1987-07-05
526	526	XJ	Female	1997-07-03
527	527	SS	Female	1997-04-02
528	528	RV	Male	1968-12-02
529	529	OX	Male	1979-05-06
530	530	JJ	Female	1966-02-07

Лістинг для таблиці doctor:

```
INSERT INTO doctor (doctor_id, s_f_p, specialization)

SELECT

    generate_series(1, 100000) AS doctor_id,

    chr(65 + trunc(random()*25)::int) || chr(65 +
trunc(random()*25)::int) AS s_f_p,

    (ARRAY['Cardiology', 'Neurology', 'Orthopedics', 'Pediatrics',
'Dermatology'])[trunc(random() * 5 + 1)::int] AS specialization
```

Лістинг для таблиці patient:

```
INSERT INTO patient (patient_id, s_f_p, gender, DOB)

SELECT

    generate_series(1, 100000) AS patient_id,

    chr(65 + trunc(random()*25)::int) || chr(65 +
trunc(random()*25)::int) AS s_f_p,

    (ARRAY['Male', 'Female'])[trunc(random() * 2 + 1)::int] AS
gender,

    date '1950-01-01' + trunc(random() * 18250)::int AS DOB --
Random date between 1950 and 2000
```

Пошук медичної історії за ід лікаря та пацієнта

```
Please make a selection: 8
Enter doctor ID: 1
Enter patient ID: 1
medical_history_id: 1 diagnos_name: gregregre description: ergre doctor_s_f_p: BU patient_s_f_p: FC
```

```
SELECT mh.medical_history_id, mh.diagnos_name, mh.description,
d.s_f_p AS doctor_name, p.s_f_p AS patient_name

FROM public.medical_history mh

JOIN public.doctor d ON mh.doctor_id = d.doctor_id

JOIN public.patient p ON mh.patient_id = p.patient_id
```

WHERE mh.doctor_id = %s AND mh.patient_id = %s;

with parameters: 1, 1

Пошук рецепту по id медичної історії(виводиться зразу назви препаратів та діагноз)

```
Please make a selection: 9
Enter medical history ID: 1
recept_id: 1 drug_name: vitamin diagnos_name: gregregre
```

SELECT r.recept_id, r.drug_name, mh.diagnos_name

FROM recept r

JOIN medical_history mh ON r.medical_history_id =
mh.medical_history_id

WHERE r.medical_history_id = %s;

with parameter: 1

Час виконання запиту: 0.0013 секунд

Короткий опис функцій модуля:

__init__ - Ініціалізує з'єднання з базою даних і налаштовує логування.

execute_sql - Виконує довільний SQL-запит.

get_all_tables - Повертає список усіх таблиць у схемі public.

get_all_columns - Повертає список колонок для вказаної таблиці.

get_all_column_types - Отримує типи даних для заданих колонок таблиці.

get_foreign_keys - Повертає зовнішні ключі таблиці.

add_data - Додає нові записи в таблицю.

generate_doctor_data - Генерує тестові дані для таблиці doctor.

generate_patient_data - Генерує тестові дані для таблиці patient.

generate_medical_history_data - Генерує тестові дані для таблиці medical_history.

generate_recept_data - Генерує тестові дані для таблиці recept.

generate_data - Викликає генерацію даних для однієї з таблиць.

read_data - Читає всі дані з вказаної таблиці.

delete_data - Видаляє запис із таблиці за ідентифікатором.

update_data - Оновлює дані в таблиці за ідентифікатором.

find_medical_history - Повертає медичну історію за ID лікаря та пацієнта.

find_recepts_by_medical_history - Повертає рецепти за ID медичної історії.

close_connection - Закриває з'єднання з базою даних.